สุทธิศักดิ์ พงศ์ธนาพาณิช

SOFT MAIL

เปิดโลกมัลติมีเดียด้วย Visual BASIC ตอน การสร้าง Screen Saver ตอนจบ

เทคนิคการแสดงผลภาพกราฟิก
โดยมีลักษณะการแสดงผลแบบต่อเนื่อง

จากบทความตอนที่แล้วผู้เขียนได้เขียนถึงเทคนิคการสร้างโปรแกรม Screen Saver สำหรับวินโดวส์ 95 ด้วย Visual BASIC 5.0 โดยได้แสดงถึงรายละเอียดทั้งหมดที่เป็นเทคนิคที่สำคัญ สำหรับการสร้างโปรแกรม Screen Saver ซึ่งผู้อ่านสามารถนำไปปรับปรุงโดยการเพิ่มเติมความสามารถในการสร้างภาพกราฟิกต่าง ๆ ได้ตามความเหมาะสม และสำหรับในบทความตอนนี้ผู้เขียนก็จะขอเพิ่มเติมเทคนิคในการแสดงผลภาพกราฟิกซึ่งมีลักษณะที่แตกต่างจากตอนที่ผ่านมาโดยมีลักษณะการแสดงผลแบบต่อเนื่อง แต่ยังคงอาศัยคอนโทรล Timer เช่นเดิม ดังนั้นในบทความตอนนี้จึงถือได้ว่าจะมีประโยชน์อย่างมากกับผู้อ่านที่จะสามารถนำไปปรับปรุงเพิ่มความสามารถได้ตามที่ต้องการ

แต่ก่อนที่ผู้เขียนจะได้กล่าวถึงรายละเอียด ก็จะขอเตือนให้ผู้อ่านที่มีไฟล์โปรแกรม Screen Saver ของตอนที่ผ่านมาในเครื่องอยู่แล้ว ให้ระวังการเขียนทับไฟล์เดิม เนื่องจากโปรแกรมตัวอย่างสำหรับบทความตอนนี้จะมีชื่อเหมือนกับไฟล์ทั้งหมดของโปรแกรมตัวอย่างของตอนที่ผ่านมา ดังนั้นในการคัดลอกซอร์สโค้ดผู้อ่านก็ควรที่จะคัดลอกลงในไดเรคทอรีที่แยกออกจากไดเรคทอรีที่จัดเก็บซอร์สโค้ดไฟล์โปรแกรม Screen Saver ตอนที่ผ่านมา เพื่อป้องกันการเขียนข้อมูลทับไฟล์เดิม

เทคนิคการแสดงผลกราฟิกแบบต่อเนื่อง

จากบทความเทคนิคการสร้างโปรแกรม Screen Saver ตอนที่แล้ว ผู้อ่านจะเห็นว่าผู้เขียนได้ทำการสร้างโค้ดสำหรับการแสดงผลกราฟิกทั้งหมดเอาไว้ในโพรซีเดอร์เหตุการณ์ Timer1_Timer ซึ่งโพรซีเดอร์เหตุการณ์ดังกล่าวจะถูกกระตุ้นอย่างอัตโนมัติโดย Visual BASIC ตามช่วงเวลาที่กำหนดให้กับคุณสมบัติ Interval และโค้ดสำหรับการแสดงผลกราฟิกดังกล่าว จะเป็นการแสดงผลกราฟิกแบบสุ่ม ซึ่งในแต่ละครั้งที่โพรซีเดอร์เหตุการณ์ Timer1_Timer ถูกกระตุ้น กราฟิกที่ถูกสร้างในครั้งที่ผ่านมาก็จะไม่มีความสัมพันธ์กับกราฟิกที่ถูกสร้างในฉัด ๆ ไป ดังนั้นจึงเป็นเรื่องง่ายที่จะเขียนโค้ด แต่สำหรับในบทความตอนนี้ผู้เขียนจะแสดงเทคนิคการเขียนโค้ดเพื่อแสดงผลกราฟิกแบบต่อเนื่อง โดยจะยังคงเขียนโค้ดทั้งหมดลงในโพรซีเดอร์เหตุการณ์ Timer1_Timer แต่ทุกครั้งที่โพรซีเดอร์เหตุการณ์ Timer1_Timer ถูกกระตุ้น โค้ดดังกล่าวก็จะทำการแสดงผลกราฟิกอย่างต่อเนื่อง โดยกราฟิกที่จะแสดงในส่วนต่อ ๆ ไปก็จะมีความสัมพันธ์กับกราฟิกส่วนที่ผ่านมา ดังนั้นโค้ดในส่วนนี้จึงต้องคำนึงถึงความสัมพันธ์ดังกล่าว

แต่ก่อนที่ผู้เขียนจะกล่าวถึงรายละเอียดต่อไป ก็ขอย้อนกลับไปถึงโครงสร้างการทำงานหลักของโปรแกรม Screen Saver เสียก่อน เนื่องจากโครงสร้างการทำงานดังกล่าว ยังถือว่าเป็นส่วนหลักของการเขียนโค้ดสำหรับสร้างโปรแกรม Screen Saver ดังในโฟล์วชาร์ตรูปที่ 1 ดังนี้

1. การกำหนดค่าเริ่มต้นต่าง ๆ สำหรับใช้ในการสร้างภาพและเป็นเงื่อนไขในการทำงานของ Screen Saver ซึ่งในโปรแกรมตัวอย่างผู้เขียนจะใช้วิธีการอ่านค่าพารามิเตอร์จากไฟล์ registry ของวินโดวส์

2. ทำการตรวจสอบการกดคีย์หรือเลื่อนเมาส์ ซึ่งการเกิดเหตุการณ์อย่างใดอย่างหนึ่ง ก็จะเป็นการจบการทำงานของ Screen Saver โดยการข้ามไปยังขั้นตอนที่ 4

3. ทำการสร้างหรือแสดงผลภาพโดยการกระตุ้นโพรซีเดอร์เหตุการณ์ Timer1_Timer และวนลูปไปยังขั้นตอนที่ 2 เพื่อการตรวจสอบต่อไป

4. หยุดกระตุ้นโพรซีเดอร์เหตุการณ์ Timer1_Timer และกำหนดให้จบการทำงานของ Screen Saver โดยการเรียกโพรซีเดอร์ที่ทำหน้าที่จบการทำงานของ Screen Saver

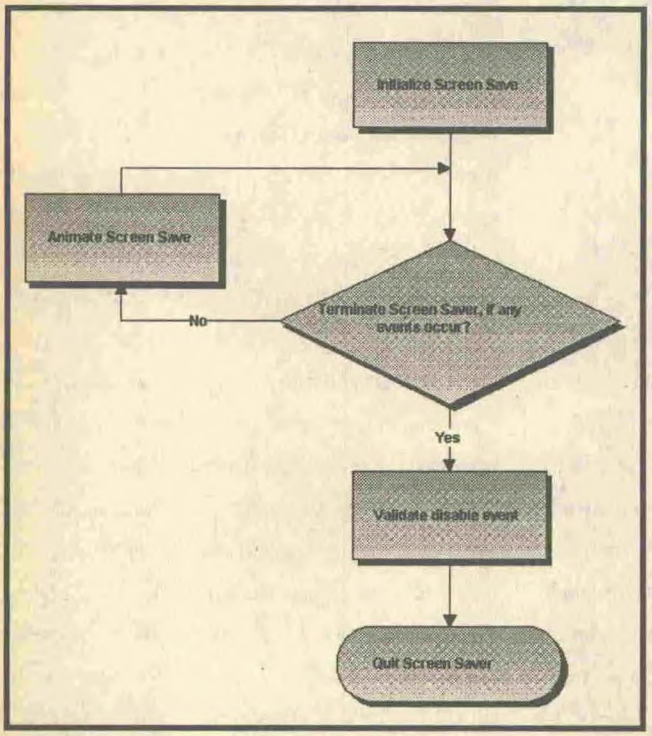
5. จบการทำงานของ Screen Saver

สำหรับขั้นตอนทั้ง 5 ลำดับข้างต้น ก็จะเป็นขั้นตอนหลักของโปรแกรม Screen Saver ซึ่งในบทความตอนนี้ ผู้เขียนได้ทำการปรับปรุงโค้ดหลัก ๆ เฉพาะในส่วนของขั้นตอนที่ 3 ซึ่งเป็นขั้นตอนการสร้างหรือแสดงผลกราฟิก สำหรับโปรแกรมตัวอย่างในบทความนี้ผู้เขียนจะขอเสนอเทคนิคการแสดงผลกราฟิกในลักษณะครั้งละ 1 บล็อก โดยจะแสดงผลกราฟิกจากซ้ายมาขวาและเมื่อสิ้นสุดแต่ละบรรทัด ก็จะแสดงผลที่บรรทัดข้างล่างถัดมา โดยผู้เขียนได้สรุปเอาไว้ในรูปแบบของโฟล์วชาร์ตดังในรูปที่ 2 ซึ่งมีลำดับขั้นตอนการเขียนโค้ดทั้งหมดนี้จะถูกเขียนลงในโพรซีเดอร์เหตุการณ์ Timer1_Timer ดังนี้

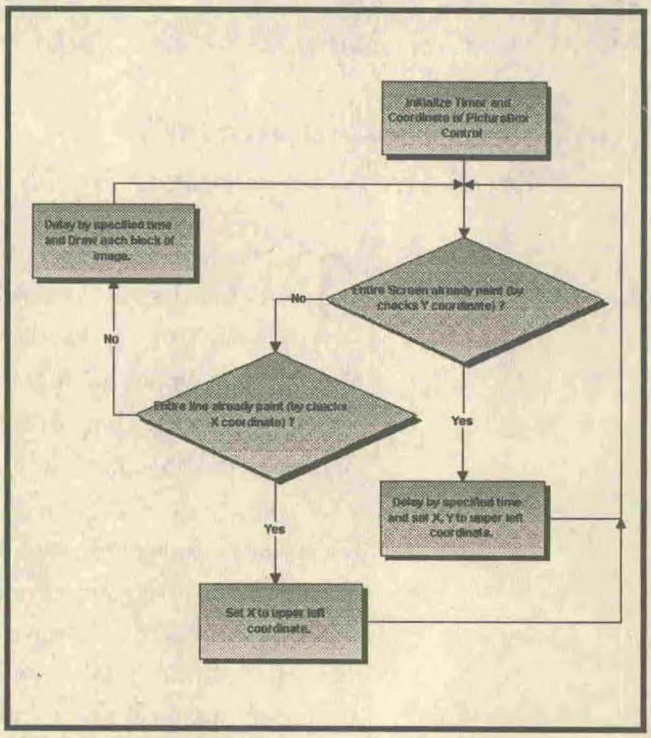
1. กำหนดค่าเริ่มต้นพารามิเตอร์สำหรับการแสดงผลกราฟิก สำหรับในโค้ดตัวอย่างของบทความนี้ ผู้เขียนจะใช้วิธีการอ่านค่าพารามิเตอร์จากไฟล์ registry ของวินโดวส์ และถ้าหากค่าพารามิเตอร์ไม่เคยถูกจัดเก็บในไฟล์ registry มาก่อน ก็จะอ่านจากค่าคงที่ปกติ (ค่าคงที่ถูกประกาศในส่วนประกาศของโมดูล ScrnSave.bas ซึ่งชื่อค่าคงที่จะขึ้นต้นด้วย DEFAULT_)

2. ตรวจสอบดูว่า บรรทัดที่แสดงผลเป็นบรรทัดถัดไปจากบรรทัดสุดท้ายใช่หรือไม่ (เพื่อยืนยันการสิ้นสุดการแสดงผลกราฟิกทั้งรูป)

- **ใช่** โดยถ้าหากค่าตัวแปร StepY มากกว่าค่าโคออร์ดิเนต Y ของมุมล่างขวาของฟอร์ม Form1 ก็แสดงว่าสิ้นสุดการแสดงผลกราฟิก ก็ให้ทำการรีเซตค่าพารามิเตอร์เริ่มต้นให้กับตัวแปร StepX และ StepY โดยให้มีค่าเท่ากับ โคออร์ดิเนต X และ Y ของมุมบนซ้ายของฟอร์ม Form1 ตามลำดับ และกำหนดให้โปรแกรมรอคอยเป็นเวลานานเท่ากับค่าที่กำหนดให้กับตัวแปร MainDelay จากนั้นก็ให้ทำการเริ่มต้นแสดงผลกราฟิกใหม่อีกครั้งตามขั้นตอนที่ 4



รูปที่ 1 แสดงขั้นตอนการทำงานหลักของโปรแกรม Screen Saver



รูปที่ 2 แสดงโครงสร้างการทำงานของ Screen Saver

- **ไม่ใช่** ให้ดำเนินการตามขั้นตอนที่ 3

3. ตรวจสอบดูว่า บล็อกที่แสดงผลในแต่ละบรรทัดเป็นบล็อกถัดไปจากบล็อกสุดท้ายใช่หรือไม่ (เพื่อยืนยันการสิ้นสุดการแสดงผลกราฟิกในแต่ละบรรทัด)

- **ใช่** โดยถ้าหากค่าตัวแปร StepX มากกว่าค่าโคออร์ดิเนต X ของมุมล่างขวาของฟอร์ม Form1 ก็แสดงว่าสิ้นสุดการแสดงผลกราฟิกในแต่ละบรรทัด ก็ให้ทำการรีเซตค่าพารามิเตอร์เริ่มต้นให้กับตัวแปร StepX โดยให้มีค่าเท่ากับ โคออร์ดิเนต X ของมุมบนซ้ายของฟอร์ม Form1 และเพิ่มค่าให้กับตัวแปร StepY เท่ากับขนาดความสูงของบล็อก เพื่อกำหนดให้แสดงผลกราฟิกบรรทัดถัดไป จากนั้นก็ให้ทำการแสดงผลกราฟิกต่อไปตามขั้นตอนที่ 4

- **ไม่ใช่** ให้ดำเนินการตามขั้นตอนที่ 4

4. เพิ่มค่าให้กับตัวแปร StepX เท่ากับขนาดความกว้างของบล็อก เพื่อกำหนดให้แสดงผลกราฟิกบล็อกถัดไป จากนั้นก็ให้ทำการแสดงผลกราฟิก

5. วนซ้ำในขั้นตอนที่ 2 จนกว่าจะมีการสิ้นสุดการทำงานของโปรแกรม Screen Saver

จากขั้นตอนข้างต้นผู้เขียนได้นำมาสร้างเป็นโปรแกรม Screen Saver โดยในขั้นตอนพื้นฐานนี้ผู้เขียนได้แบ่งโพรซีเดอร์ต่าง ๆ ใน 5 ขั้นตอนข้างต้นลงใน 3 ไฟล์หลัก ๆ เช่นเดิม ดังนี้

ไฟล์ ScrnSave.bas ประกอบด้วย โพรซีเดอร์สำหรับการเริ่มต้น (Sub) และอื่น ๆ

ไฟล์ ScrnSave.frm ประกอบด้วย โพรซีเดอร์สำหรับการสร้างภาพกราฟิก และการตรวจสอบการสิ้นสุดการทำงาน

ไฟล์ Setup.frm ประกอบด้วย โพรซีเดอร์สำหรับการกำหนดเงื่อนไขพารามิเตอร์ของโปรแกรม

หลักการทำงานข้างต้นทั้งหมด ก็จะแสดงให้เห็นถึงขั้นตอนที่สำคัญ ๆ ทั้งหมด ในการแสดงผลกราฟิกแบบต่อเนื่องสำหรับโปรแกรมประเภท Screen Saver โดยที่เราจะเขียนโค้ดสำหรับการแสดงผลกราฟิกทั้งหมดเอาไว้ในโพรซีเดอร์เหตุการณ์ Timer1_Timer ทั้งนี้เพื่อให้ง่ายต่อการตรวจสอบข้อผิดพลาดของการทำงานของโค้ด สำหรับในหัวข้อถัดไปผู้เขียนก็จะทำการอธิบายถึงการทำงานของโค้ดทั้งหมดของโปรแกรมตัวอย่าง ScrnSave.vbp ต่อไป

การทำงานของโปรเจ็กต์ ScrnSave.vbp

เมื่อมีการรันโปรแกรม โพรซีเดอร์ Main ก็จะถูกเรียกเป็นโพรซีเดอร์แรก เนื่องจากผู้เขียนได้กำหนดให้เป็นโพรซีเดอร์เริ่มต้นของโปรแกรม ซึ่งจะเริ่มต้นด้วยการตรวจสอบ instance ของโปรแกรม Screen Saver ว่ามีการทำงานอยู่แล้วก่อนหน้านี้หรือไม่ โดยถ้าหากมีก็ให้สิ้นสุดการทำงาน ดังนี้

```
If App.PrevInstance Then End
```

จากนั้นก็ทำการอ่านค่าเริ่มต้นของพารามิเตอร์ ซึ่งได้ถูกกำหนดให้จัดเก็บเอาไว้ในไฟล์ registry ของวินโดวส์ ภายใต้ชื่อแอพพลิเคชัน SEED_P_SUTTHISAK และส่วน ScrSaver02 ซึ่งจะประกอบด้วย 5 พารามิเตอร์ สำหรับการกำหนดเงื่อนไขในการแสดงกราฟิกเป็นบล็อกอย่างต่อเนื่อง ดังนี้

ส่วนคำสั่ง StepWidth จัดเก็บค่าขนาดความกว้างของบล็อกสำหรับแสดงผลกราฟิก ในหน่วยของจุด (pixel)

ส่วนคำสั่ง StepHeight จัดเก็บค่าขนาดความสูงของบล็อกสำหรับแสดงผลกราฟิก ในหน่วยของจุด (pixel)

ส่วนคำสั่ง StepDelay จัดเก็บค่าเวลาที่ใช้หน่วง ในระหว่างการแสดงผลกราฟิกแต่ละบล็อก ในหน่วยของมิลลิวินาที

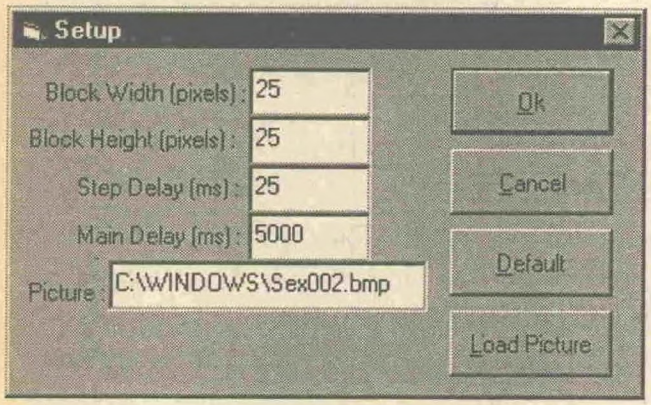
ส่วนคำสั่ง MainDelay จัดเก็บค่าเวลาที่ใช้หน่วง เมื่อกราฟิกทั้งรูปได้ถูกแสดงผลอย่างสมบูรณ์ ก่อนที่จะทำการลบทิ้งจอภาพเพื่อเริ่มต้นแสดงผลกราฟิกใหม่อีกครั้ง ในหน่วยของมิลลิวินาที

ส่วนคำสั่ง BackgroundPicture จัดเก็บชื่อของไดเรคทอรีและชื่อไฟล์กราฟิก เช่น .bmp, .gif, .jpg เป็นต้น สำหรับใช้เป็นกราฟิกต้นทางสำหรับนำไปแสดงผลครั้งละ 1 บล็อก เมื่อ Screen Saver ทำงาน

ซึ่งการอ่านค่าพารามิเตอร์จากไฟล์ registry ของวินโดวส์ ก็สามารถกระทำได้โดยใช้ฟังก์ชัน GetSetting ของ Visual BASIC ดังนี้

```
appNameStr$ = "SEED_P_SUTTHISAK"
sectionStr$ = "ScrSaver02"
ScrnCo.Wdt = GetSetting(appNameStr$, sectionStr$, "StepWidth", _
        CStr(DEFAULT_STEPWIDTH))
ScrnCo.Hgt = GetSetting(appNameStr$, sectionStr$, "StepHeight", _
        CStr(DEFAULT_STEPHEIGHT))
ScrnCo.StepDelay = GetSetting(appNameStr$, sectionStr$, "StepDelay", _
        CStr(DEFAULT_STEPDELAY))
ScrnCo.MainDelay = GetSetting(appNameStr$, sectionStr$, "MainDelay", _
        CStr(DEFAULT_MAINDELAY))
BkgPic = GetSetting(appNameStr$, sectionStr$, _
        "BackgroundPicture", CStr(""))
```

และเนื่องจากการเรียกโปรแกรมประเภท Screen Saver โดยวินโดวส์นั้น สามารถเป็นไปได้ใน 2 กรณี คือ การเรียกเพื่อรัน Screen Saver หรือเพื่อการ setup พารามิเตอร์ ดังนั้นจึงต้องมีการตรวจสอบเงื่อนไขในการเรียกโปรแกรม โดยที่วินโดวส์จะกำหนดให้ใช้ command line "/C" ในการเรียกโปรแกรมเพื่อเป็นการ setup พารามิเตอร์ ดังนั้นถ้าหากวินโดวส์เรียกโปรแกรม Screen Saver โดยมี command line "/C" (ซึ่งวินโดวส์จะกระทำโดยอัตโนมัติ เมื่อผู้อ่านคลิกที่ปุ่ม Setting... ของรายการ Display Properties ของ Control Panel) ก็จะทำการแสดงผล



รูปที่ 3 แสดงฟอร์มสำหรับการ setup ค่าพารามิเตอร์ของ Screen Saver

ฟอร์ม Form2 เพื่อแสดงฟอร์มสำหรับการกำหนดค่าพารามิเตอร์ และให้สิ้นสุดการทำงานของโปรแกรมหลังจากจบการกำหนดค่าพารามิเตอร์ ดังนี้

```
If UCase$(Command$) = "/C" Then
 Form2.Show vbModal
 End
End If
```

ซึ่งฟอร์ม Form2 ก็จะถูกแสดงผลดังในรูปที่ 3 โดยภายในฟอร์มก็จะอนุญาตให้ผู้อ่านสามารถกำหนดค่าพารามิเตอร์ทั้ง 5 ตามที่กล่าวมาแล้วข้างต้น และถ้าหากผู้อ่านต้องการให้กำหนดค่า default สำหรับพารามิเตอร์ทั้งหมด ก็ให้คลิกที่ปุ่มคำสั่ง Default

แต่ถ้าหากวินโดวส์เรียก Screen Saver เพื่อทำงานโดยตรง ก็ให้ทำงานต่อโดยการเรียกโพรซีเดอร์ HideCursor เพื่อซ่อนเมาส์ โดยใช้เทคนิคการเลื่อนตำแหน่งของเมาส์ไปยังมุมล่าง-ขวาของจอภาพ โดยอาศัยฟังก์ชัน API ของวินโดว์ SetCursorPos และซ่อนเมาส์โดยอาศัยฟังก์ชัน API ของวินโดว์ ShowCursor และสุดท้ายก็จัดเก็บตำแหน่งของเมาส์ก่อนการย้ายสำหรับนำมาใช้คืนกลับตำแหน่งของเมาส์เมื่อสิ้นสุดการทำงานของโปรแกรม ดังนี้

```
XPos% = Screen.Width
YPos% = Screen.Height

SetCursorPos XPos%, YPos%
Result& = ShowCursor(-1)

LastX! = XPos%
LastY! = YPos%
```

และแสดงผล Form1 ซึ่งเป็นฟอร์มหลักสำหรับการสร้างภาพกราฟิกของ Screen Saver

```
Form1.Show
```

เมื่อฟอร์ม Form1 แสดงผล โพรซีเดอร์ Form_Load ก็จะถูกกระตุ้น โดยภายในโพรซีเดอร์ก็จะเป็นการกำหนดขนาดของฟอร์มให้เต็มจอภาพและกำหนดให้พื้นหลังของฟอร์มเป็นสีดำ จากนั้นก็ทำการโหลดไฟล์กราฟิกให้กับคอนโทรล Picture1 ซึ่งถูกใช้เป็นแหล่งต้นทางในการจัดเก็บกราฟิก และกำหนดค่าของ Timer1 ให้เท่ากับค่าของตัวแปร ScrnCo.StepDelay ดังนี้

```
Me.WindowState = vbMaximized
BackColor = QBColor(0)
FirstTime = True

' Load background picture.
Picture1.Picture = LoadPicture(BkgPic)

Timer1.Interval = ScrnCo.StepDelay
```

และสำหรับโพรซีเดอร์อื่น ๆ ผู้เขียนก็ไม่ได้ทำการดัดแปลงแต่ประการใด ดังนั้นจึงไม่ขอกล่าวรายละเอียดในบทความตอนนี้ แต่ถ้าหากผู้อ่านท่านใดสนใจก็ขอให้ติดตามอ่านได้จากบทความในตอนที่ผ่านมาครับ ส่วนโพรซีเดอร์ที่มีการแก้ไขโค้ดทั้งหมดและมีความสำคัญมากที่สุดก็คือโพรซีเดอร์เหตุการณ์ Timer1_Timer ซึ่งมีรายละเอียดของการทำงานดังนี้

เริ่มต้นด้วยการตรวจสอบค่าตัวแปร stepY กับ ScrnCo.lrY เพื่อดูว่าการแสดงผลกราฟิกได้เสร็จสิ้นสมบูรณ์แล้วหรือยัง โดยถ้าหากค่า stepY มากกว่าหรือเท่ากับ ScrnCo.lrY ก็แสดงว่าการแสดงผลกราฟิกเสร็จสมบูรณ์ ก็ให้ทำการลบทิ้งกราฟิกจากฟอร์ม Form1 จากนั้นกำหนดค่าเริ่มต้นเกี่ยวกับโคออร์ดิเนตให้กับตัวแปรที่เกี่ยวข้องใหม่อีกครั้ง ดังนี้

```
If stepY >= ScrnCo.lrY Then
 Me.Cls

 Picture1.ScaleHeight = Me.ScaleHeight
 Picture1.ScaleWidth = Me.ScaleWidth
 ScrnCo.ulX = Me.ScaleLeft
 ScrnCo.ulY = Me.ScaleTop
 ScrnCo.lrX = Me.ScaleWidth
 ScrnCo.lrY = Me.ScaleHeight

 stepX = ScrnCo.ulX
 stepY = ScrnCo.ulY

 DelayTime ScrnCo.MainDelay
```

แต่ถ้าหากค่า stepY น้อยกว่า ScrnCo.lrY ก็ให้ทำการตรวจสอบดูว่าค่า stepX มากกว่าหรือเท่ากับ ScrnCo.lrX หรือไม่ โดยถ้าหากค่า stepX มากกว่าหรือเท่ากับ ScrnCo.lrX ก็แสดงว่าการแสดงผลกราฟิกในบรรทัดปัจจุบันเสร็จสมบูรณ์ ก็ให้ทำการลบทิ้งรีเซตค่าของ stepX และบวกค่าความสูงของบล็อก (ตัวแปร ScrnCo.Hgt) ให้กับตัวแปร stepY เพื่อวาดกราฟิกบรรทัดใหม่ ดังนี้

```
 ElseIf stepX >= ScrnCo.lrX Then
 stepX = ScrnCo.ulX
 stepY = stepY + ScrnCo.Hgt
```

แต่ถ้าหากไม่เป็นจริงทั้งสองกรณีข้างต้น ก็แสดงว่าอยู่ในระหว่างขั้นตอนการวาดบล็อกในบรรทัดปัจจุบัน ก็ให้ทำการวาดบล็อกของกราฟิกด้วยประโยค PaintPicture จากนั้นก็ขยับไปยังบล็อกถัดไป ด้วยการบวกค่าความกว้างของบล็อกให้กับตัวแปร stepX และสุดท้ายก็ให้ทำการหน่วงเวลาก่อนที่จะแสดงผลกราฟิกบล็อกต่อไปด้วยการเรียกโพรซีเดอร์ DelayTime ดังนี้

```
Else
  PaintPicture Picture1.Picture, stepX, stepY,_
   ScrnCo.Wdt,ScrnCo.Hgt, stepX, stepY,  _
   ScrnCo.Wdt, ScrnCo.Hgt
  stepX = stepX + ScrnCo.Wdt
  DelayTime ScrnCo.StepDelay
End If
```

สำหรับขบวนการวาดรูปวงกลมนี้จะดำเนินต่อไปเรื่อย ๆ จนกว่าผู้อ่านจะมีการเลื่อนเมาส์หรือกดคีย์ใด ๆ บนคีย์บอร์ด ก็จะมีการเขียนโค้ดภายในโพรซีเดอร์ Form_MouseDown, Form_MouseMove และ Form_KeyDown ตามลำดับ เพื่อทำการตรวจสอบเงื่อนไขเมื่อการสิ้นสุดการทำงานของโปรแกรม โดยการเรียกใช้โพรซีเดอร์ MonitorEvents ซึ่งจะมีการตรวจสอบโดยเงื่อนไข โดยถ้าหากโพรซีเดอร์เกี่ยวกับการกดคีย์บอร์ดถูกเรียก ก็ให้สิ้นสุดการทำงานของโปรแกรม โดยการเรียกใช้โพรซีเดอร์ QuitScreenSaver แต่ถ้าหากใช่ก็ให้ทำการหน่วงเวลา 1 วินาที เพื่อป้องกันการเรียกโพรซีเดอร์ข้างต้นซ้ำโดยวินโดวส์ และกำหนดให้ FirstTime เท่ากับ False ดังนี้

```
If (Not FirstTime) Or LastX! = KEYDOWNEXIT Then
  Timer1.Enabled = False
  QuitScreenSaver
Else
   ' Trap extra mouse move events
   ' Delay before set FirstTime to be False
   ' is very important. Please try to
   ' cancel this statement and re-run again
  DelayTime 1000
  FirstTime = False
End If
```

สำหรับขั้นตอนข้างต้นทั้งหมด ก็จะเป็นขั้นตอนหลัก ๆ สำหรับการเขียนโค้ดเพื่อสร้างโปรแกรมประเภท Screen Saver สำหรับวินโดวส์ สำหรับขั้นตอนสุดท้ายก่อนที่จะสามารถนำโปรแกรมประเภท Screen Saver ไปใช้งาน ก็คือการสร้างหรือคอมไพล์ไฟล์ดังกล่าวเป็นไฟล์ .exe โดยการเลือกคำสั่ง Make .EXE ของเมนู File ของ Visual BASIC 5.0 ดังที่ได้กล่าวมาแล้วในบทความตอนที่ผ่านมา ซึ่งจากเนื้อหาทั้งหมดในบทความตอนนี้ผู้เขียนหวังว่าผู้อ่านคงจะได้เห็นแนวทางในการปรับปรุงขีดความสามารถในการแสดงผลกราฟิกของโปรแกรมประเภท Screen Saver ซึ่งผู้อ่านสามารถนำไปเพิ่มเติมโค้ดสำหรับการแสดงผลลูกเล่นต่าง ๆ ได้ตามที่ต้องการต่อไป พบกันใหม่ในตอนต่อไปสวัสดีครับ

ลิสติงที่ 1
ไฟล์โปรเจ็กต์
ScrnSave.vbp

```
Type=Exe
Module=basScreenSaver; ScrnSave.bas
Form=ScrnSave.Frm
Form=Setup.Frm
Reference=*\G{00020430-0000-0000-C000-000000000046}#2.0#0# _
 C:\WINDOWS\SYSTEM\STDOLE2.TLB#Standard OLE Types
Reference=*\G{EE008642-64A8-11CE-920F-08002B369A33}#1.0#0# _
 C:\WINDOWS\SYSTEM\MSRDO20.DLL#Microsoft Remote Data Object 1.0
Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.1#0; COMDLG32.OCX
Startup="Sub Main"
HelpFile=""
Title="SCRNSAVE:SEED-ScreenSaver02"
ExeName32="ScrnSave.scr"
Path32="c:\windows\system"
Command32=""
Name="ScreenSaver02"
HelpContextID="0"
CompatibleMode="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionComments="Program for Microcomputer Magazine."
VersionCompanyName="SE-ED Co.,Ltd."
VersionFileDescription="Screen Saver 02"
VersionLegalCopyright="Sutthisak Phongthanapanich"
VersionProductName="Multimedia Programming Technique"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
ThreadPerObject=0
MaxNumberOfThreads=1
```

ลิสติงที่ 2
ไฟล์โมดูล
ScrnSave.bas

```
Attribute VB_Name = "basScreenSaver"
Option Explicit
'
' Project name : ScrnSave.vbp
' Program Name : Screen Saver-02 Demonstration
' VB/Win Module : ScrnSave.bas
' DLL filename : -
' Copyright : (c)1998 by Sutthisak Phongthanapanich
'

' Default constants.
Public Const DEFAULT_STEPWIDTH = 25
Public Const DEFAULT_STEPHEIGHT = 25
Public Const DEFAULT_STEPDELAY = 25
Public Const DEFAULT_MAINDELAY = 5000

Public Type ScrnCoType
 ulX As Long
 ulY As Long
 lrX As Long
 lrY As Long
 Hgt As Long
 Wdt As Long
 StepDelay As Long
 MainDelay As Long
End Type

Public ScrnCo As ScrnCoType
Public BkgPic As String

' Windows API declaration.
Public Declare Function ShowCursor Lib "user32" (ByVal fShow As Long) As Long
Public Declare Function SetCursorPos Lib "user32" (ByVal x As Long, _
 ByVal y As Long) As Long

' Used to mark keyboard events
Public Const KEYDOWNEXIT = -1

' Circles screen saver setup related values
Public appNameStr$, sectionStr$, keystr$

' Public variable used to track mouse movements
Public FirstTime As Boolean
Public LastX!, LastY!

Sub Main()
 ' Stop running program if there's already a copy of running
 If App.PrevInstance Then End

 ' Initialize application variables by read from registry
 appNameStr$ = "SEED_P_SUTTHISAK"
 sectionStr$ = "ScrSaver02"
 ScrnCo.Wdt = GetSetting(appNameStr$, sectionStr$, "StepWidth", _
 CStr(DEFAULT_STEPWIDTH))
 ScrnCo.Hgt = GetSetting(appNameStr$, sectionStr$, "StepHeight", _
 CStr(DEFAULT_STEPHEIGHT))
 ScrnCo.StepDelay = GetSetting(appNameStr$, sectionStr$, "StepDelay", _
 CStr(DEFAULT_STEPDELAY))
 ScrnCo.MainDelay = GetSetting(appNameStr$, sectionStr$, "MainDelay", _
 CStr(DEFAULT_MAINDELAY))
 BkgPic = GetSetting(appNameStr$, sectionStr$, "BackgroundPicture", CStr(""))

 ' Configure screen saver instead of running it
 ' Be sure to initialize setup with values you retrieve from profile file
 If UCase$(Command$) = "/C" Then
 Form2.Show vbModal
 DelayTime 1000
 End
 End If

 ' Move cursor off-screen
 HideCursor
 ' Start screen saver graphics form
 Form1.Show
End Sub

Public Sub CenterForm(F As Form)
 Dim cTop%, cLeft%

 ' Calculate top-left coordination
 cTop% = (Screen.Height - F.Height) \ 2
 cLeft% = (Screen.Width - F.Width) \ 2
 If F.WindowState = vbNormal Then
 If cTop% < 0 Then F.Top = 0 Else F.Top = cTop%
 If cLeft% < 0 Then F.Left = 0 Else F.Left = cLeft%
 End If
End Sub

Function GetRandom%(FromNum%, ToNum%)
 ' Returns a random integer between BegNbr% and EndNbr% inclusive
 GetRandom% = Int((ToNum% - FromNum% + 1) * Rnd + FromNum%)
End Function

' Set cursor screen position values to lower right corner
Public Sub HideCursor()
```

→

เปิดโลกมัลติมีเดียด้วย Visual BASIC

ตอน การสร้าง Screen Saver ตอนจบ

ลิสติงที่ 2 (ต่อ)

```
    Dim Result&, XPos%, YPos%

    ' Get bottom-right screen coordination
    XPos% = Screen.Width
    YPos% = Screen.Height
    ' Set cursor position
    SetCursorPos XPos%, YPos%
    Result& = ShowCursor(-1)

    ' Reset values for eliminating system jiggle
    LastX! = XPos%
    LastY! = YPos%
   End Sub

   Public Sub UnhideCursor()
    Dim Result&
    ' restore original cursor position
    SetCursorPos LastX!, LastY!
    Result& = ShowCursor(0)
   End Sub

   Public Sub DelayTime(ByVal msTime As Long)
    Dim BeginTime&

    ' Delay time in specified period (ms).
    BeginTime& = Timer
    Do Until Abs(Timer - BeginTime&) >= (msTime / 1000)
    Loop
   End Sub
```

■

ลิสติงที่ 3 ไฟล์ฟอร์ม ScrnSave.frm

```
VERSION 5.00
Begin VB.Form Form1
 AutoRedraw = -1 'True
 ClientHeight = 1335
 ClientLeft = 1170
 ClientTop = 1515
 ClientWidth = 2685
 ControlBox = 0 'False
 FillStyle = 0 'Solid
 LinkTopic = "Form1"
 MaxButton = 0 'False
 MinButton = 0 'False
 PaletteMode = 1 'UseZOrder
 ScaleHeight = 89
 ScaleMode = 3 'Pixel
 ScaleWidth = 179
 WindowState = 2 'Maximized
 Begin VB.PictureBox Picture1
 Align = 1 'Align Top
 AutoRedraw = -1 'True
 Height = 375
 Left = 0
 ScaleHeight = 21
 ScaleMode = 3 'Pixel
 ScaleWidth = 175
 TabIndex = 0
 Top = 0
 Visible = 0 'False
 Width = 2685
 End
 Begin VB.Timer Timer1
 Interval = 500
 Left = 300
 Top = 300
 End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
'
' Project name : ScrnSave.vbp
' Program Name : Screen Saver-02 Demonstration
' VB/Win Module : ScrnSave.frm
' DLL filename : -
' Copyright : (c)1998 by Sutthisak Phongthanapanich
'

Sub Form_Load()
 ' Initialize screen saver and maximize the window
 Me.WindowState = vbMaximized
 BackColor = QBColor(0)
 FirstTime = True

 ' Load background picture.
 Picture1.Picture = LoadPicture(BkgPic)

 ' Set timer interval tigger
 Timer1.Interval = ScrnCo.StepDelay
End Sub

Sub Form_MouseDown(Button As Integer, Shift As Integer, x As
Single, y As Single)
 MonitorEvents x, y
End Sub

Sub Form_MouseMove(Button As Integer, Shift As Integer, x As
Single, y As Single)
 MonitorEvents x, y
End Sub

Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
 MonitorEvents KEYDOWNEXIT, KEYDOWNEXIT
End Sub

Sub MonitorEvents(x As Single, y As Single)
 ' Check mouse position by compare with previously one.
 If x = LastX! And y = LastY! Then
 Exit Sub
 Else
 LastX! = x: LastY! = y
 End If

 If (Not FirstTime) Or LastX! = KEYDOWNEXIT Then
 Timer1.Enabled = False
 QuitScreenSaver
 Else
 ' Trap extra mouse move events
 ' Delay before set FirstTime to be False
 ' is very important. Please try to
 ' cancel this statement and re-run again
 DelayTime 1000
 FirstTime = False
 End If
End Sub

Sub QuitScreenSaver()
 ' Common subprocedure that all routines can call
 ' to execute any special shutdown procedures.
 UnhideCursor
 DelayTime 1000
 End
End Sub

Private Sub Timer1_Timer()
Static stepX, stepY

 If stepY >= ScrnCo.lrY Then
 Me.Cls

 Picture1.ScaleHeight = Me.ScaleHeight
 Picture1.ScaleWidth = Me.ScaleWidth
 ScrnCo.ulX = Me.ScaleLeft
 ScrnCo.ulY = Me.ScaleTop
 ScrnCo.lrX = Me.ScaleWidth
 ScrnCo.lrY = Me.ScaleHeight

 stepX = ScrnCo.ulX
 stepY = ScrnCo.ulY

 DelayTime ScrnCo.MainDelay
 ElseIf stepX >= ScrnCo.lrX Then
 stepX = ScrnCo.ulX
 stepY = stepY + ScrnCo.Hgt
 Else
 PaintPicture Picture1.Picture, stepX, stepY, ScrnCo.Wdt, _
 ScrnCo.Hgt, stepX, stepY, ScrnCo.Wdt, ScrnCo.Hgt
 stepX = stepX + ScrnCo.Wdt
 DelayTime ScrnCo.StepDelay
 End If
End Sub
```

■

ลิสติงที่ 4 ไฟล์ฟอร์ม Setup.frm

```
VERSION 5.00
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.1#0";
"COMDLG32.OCX"
Begin VB.Form Form2
 BorderStyle = 3 'Fixed Dialog
 Caption = "Setup"
 ClientHeight = 2535
 ClientLeft = 3165
 ClientTop = 2880
 ClientWidth = 4650
 LinkTopic = "Form2"
 MaxButton = 0 'False
 MinButton = 0 'False
 PaletteMode = 1 'UseZOrder
 ScaleHeight = 2535
 ScaleWidth = 4650
 ShowInTaskbar = 0 'False
 Begin MSComDlg.CommonDialog CommonDialog1
 Left = 480
 Top = 2040
 _ExtentX = 847
 _ExtentY = 847
 _Version = 327680
 CancelError = -1 'True
 DefaultExt = "*.bmp"
 DialogTitle = "Load Background Picture"
 End
 Begin VB.CommandButton Command4
 Caption = "&Load Picture"
 Height = 495
 Left = 3240
 TabIndex = 13
 Top = 1920
 Width = 1215
 End
 Begin VB.TextBox Text5
 Height = 375
 Left = 720
 Locked = -1 'True
 TabIndex = 11
 Top = 1560
 Width = 2355
 End
 Begin VB.CommandButton Command3
 Caption = "&Default"
 Height = 495
 Left = 3240
 TabIndex = 10
 Top = 1320
 Width = 1215
 End
 Begin VB.TextBox Text4
 Height = 375
 Left = 1740
 MaxLength = 4
 TabIndex = 8
 Top = 1200
 Width = 915
 End
 Begin VB.TextBox Text3
 Height = 375
 Left = 1740
 MaxLength = 4
 TabIndex = 6
 Top = 840
 Width = 915
 End
 Begin VB.CommandButton Command2
 Cancel = -1 'True
 Caption = "&Cancel"
 Height = 495
 Left = 3240
 TabIndex = 5
 Top = 720
 Width = 1215
 End
 Begin VB.CommandButton Command1
 Caption = "&Ok"
 Default = -1 'True
 Height = 495
 Left = 3240
 TabIndex = 4
 Top = 120
 Width = 1215
 End
 Begin VB.TextBox Text2
 Height = 375
 Left = 1740
 MaxLength = 3
 TabIndex = 3
 Top = 480
 Width = 915
 End
 Begin VB.TextBox Text1
 Height = 375
 Left = 1740
 MaxLength = 3
 TabIndex = 2
 Top = 120
 Width = 915
 End
 Begin VB.Label Label5
 AutoSize = -1 'True
 Caption = "Picture : "
 Height = 210
 Left = 120
 TabIndex = 12
 Top = 1680
 Width = 630
 End
 Begin VB.Label Label4
 AutoSize = -1 'True
 Caption = "Main Delay (ms) :"
 Height = 210
 Left = 480
 TabIndex = 9
 Top = 1260
 Width = 1215
 End
 Begin VB.Label Label3
 AutoSize = -1 'True
 Caption = "Step Delay (ms) :"
 Height = 210
 Left = 480
 TabIndex = 7
 Top = 900
 Width = 1200
 End
 Begin VB.Label Label2
 AutoSize = -1 'True
 Caption = "Block Height (pixels) :"
 Height = 210
 Left = 120
 TabIndex = 1
 Top = 540
 Width = 1530
 End
 Begin VB.Label Label1
 AutoSize = -1 'True
 Caption = "Block Width (pixels) : "
 Height = 210
 Left = 240
 TabIndex = 0
 Top = 200
 Width = 1530
 End
End
Attribute VB_Name = "Form2"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
'
' Project name : ScrnSave.vbp
  Program Name : Screen Saver-02 Demonstration
' VB/Win Module : Setup.frm
' DLL filename : -
' Copyright : (c)1998 by Sutthisak Phongthanapanich
'

Private Sub Command1_Click()
 ' Update information from windows registry file.
 SaveSetting appNameStr$, sectionStr$, "StepWidth", Text1.Text
 SaveSetting appNameStr$, sectionStr$, "StepHeight", Text2.Text
 SaveSetting appNameStr$, sectionStr$, "StepDelay", Text3.Text
 SaveSetting appNameStr$, sectionStr$, "MainDelay", Text4.Text
 SaveSetting appNameStr$, sectionStr$, "BackgroundPicture", Text5.Text

 ' Quit
 Unload Me
End Sub

Private Sub Command2_Click()
```

→

ลิสติงที่ 4 (ต่อ)

```
 Unload Me
End Sub

Private Sub Command3_Click()
 ' Retrieve default value.
 Text1 = CStr(DEFAULT_STEPWIDTH)
 Text2 = CStr(DEFAULT_STEPHEIGHT)
 Text3 = CStr(DEFAULT_STEPDELAY)
 Text4 = CStr(DEFAULT_MAINDELAY)
 Text5 = CStr("")
End Sub

Private Sub Command4_Click()
 On Error Resume Next

 CommonDialog1.ShowOpen
 If Err = 32755 Then Exit Sub

 Text5 = CommonDialog1.filename
End Sub

Private Sub Form_Activate()
 CommonDialog1.Filter = "Bitmap (*.bmp)|*.bmp|" & _
 "GIF (*.gif)|*.gif|TIFF (*.tif)|*.tif|" & _
 "JPEG (*.jpg)|*.jpg|Metafile (*.wmf)|*.wmf"

 ' Retrieve information from windows registry file.
 ScrnCo.Wdt = GetSetting(appNameStr$, sectionStr$, "StepWidth", _
 CStr(DEFAULT_STEPWIDTH))
 ScrnCo.Hgt = GetSetting(appNameStr$, sectionStr$, "StepHeight", _
 CStr(DEFAULT_STEPHEIGHT))
 ScrnCo.StepDelay = GetSetting(appNameStr$, sectionStr$, "StepDelay", _
 CStr(DEFAULT_STEPDELAY))
 ScrnCo.MainDelay = GetSetting(appNameStr$, sectionStr$, "MainDelay", _
 CStr(DEFAULT_MAINDELAY))
 BkgPic = GetSetting(appNameStr$, sectionStr$, _
 "BackgroundPicture", CStr(""))

 Text1 = CStr(ScrnCo.Wdt)
 Text2 = CStr(ScrnCo.Hgt)
 Text3 = CStr(ScrnCo.StepDelay)
 Text4 = CStr(ScrnCo.MainDelay)
 Text5 = CStr(BkgPic)
End Sub
```

เทคนิคประจำคอลัมน์

ถาม อธิบายข้อแตกต่างระหว่าง Track และ Session สำหรับ CD?

ตอบ Track - CD ได้มีการจัดแบ่งพื้นที่การจัดเก็บทางกายภาพออกเป็นแทร็ก (track) โดยสำหรับ Audio CD จะใช้ในการแบ่งเพลงแต่ละเพลง และสำหรับ Video CD จะใช้ในการแบ่งสายข้อมูล MPEG และสุดท้ายสำหรับ CD-ROM จะมีเพียงแทร็กเดียวเท่านั้น สำหรับการจัดเก็บข้อมูล

Session หมายถึง การเขียนข้อมูลลงใน CD ซึ่งเซสชันก็เป็นการจัดแบ่งพื้นที่การจัดเก็บทางกายภาพเช่นเดียวกับแทร็ก เพียงแต่สำหรับแต่ละเซสชันจะถูกระบุตำแหน่งเริ่มต้นด้วยโค้ดที่เรียกว่า lead-in และระบุตำแหน่งสิ้นสุดด้วยโค้ดที่เรียกว่า lead-out

สำหรับ CD ในปัจจุบันมีทั้งชนิดที่สามารถบันทึกได้เพียงเซสชันเดียวหรือหลายเซสชัน แต่สำหรับ Audio CD จะสามารถบันทึกได้แบบเซสชันเดียว (singlisession CD) เท่านั้น ส่วน CD-ROM ส่วนใหญ่มักจะสามารถบันทึกได้แบบหลายเซสชัน (multisession CD)